Vibe Check: A Personalized Recommendation Algorithm Using Sentiment-Driven Preferences

Team 20: Jack P. Lu, Aaron Montano, Kaylie Kopicki, Nicholas Fuller

Introduction

In recent years, digital platforms have become saturated with user-generated content and business listings, making the process of discovering enjoyable venues increasingly overwhelming. Today's traditional recommendation systems prioritize quantitative metrics such as ratings and popularity, often overlooking nuanced qualitative factors that shape user experiences. These systems typically fail to capture the reasons behind user preferences—such as ambiance, aesthetics, or overall "vibe", a restaurant's emotional atmosphere, leading to recommendations that may be technically relevant but emotionally disconnected from what users truly seek. This limitation reduces user satisfaction, weakens recommendation effectiveness and trust in recommendation, and restricts discovery of experiences that align with individual tastes. As seen in analogous domains like media and entertainment, there is a growing expectation for more context-aware, emotionally intelligent recommendations. Users are no longer just looking for what is popular—they want suggestions that align with how they feel and what atmosphere they seek. This emerging need underscores the importance of developing a new, sentiment-aware recommendation system that understands the "why" behind user preferences.

Problem Statement

Existing recommendation systems often overlook the qualitative reasons behind user preferences, resulting in recommendations that are accurate yet emotionally disconnected. *Vibe Check* addresses this gap by introducing a sentiment-driven algorithm that extracts qualitative insights ("vibes") like ambiance and service from user-generated reviews. By emphasizing what users truly value, *Vibe Check* enhances recommendation accuracy, transparency, and personalization, enabling users to discover venues that closely align with their preferences and expectations.

Literature Survey

The integration of behavioral analytics and big data has significantly improved recommendation systems by deepening insights into consumer preferences (Durbach & Montibeller, 2019). As social media proliferates, large-scale user-generated data now plays a central role in enhancing recommendation accuracy and personalization (Cohen & Rozenblat, 2015). While early methods—such as graph-based clustering (Zha et al., 2001), collaborative filtering (Fanca et al., 2020), and hybrid approaches (Nagarnaik & Thomas, 2015; Zaki & Hsiao, 2005)—laid foundational groundwork, they struggled with numerical bias, limited personalization, and low transparency, highlighting the need for richer qualitative inputs (Choi, 2021).

Reliability concerns in traditional sentiment analysis have been addressed by Moldovan (2021), who introduced a majority-voting framework that blends lexicon-based tools such as VADER with cloud NLP APIs (Google, Amazon, IBM, Microsoft). This hybrid approach boosted labeling accuracy and correlated strongly with user evaluations ($\rho \approx 0.70$). Moldovan also found tool-specific tendencies: VADER labeled more reviews neutral (~14%) or negative (~9.6%), whereas Google produced fewer neutral outputs (~9.5%). Opara (2022) corroborated these findings, showing significant sentiment differences across VADER, AWS, Azure, and Google Cloud ($p < .05$); AWS was most extreme, while VADER remained the most cost-effective for large-scale preprocessing.

Among the evaluated tools, VADER emerged as particularly cost-effective and efficient for preliminary sentiment analysis, highlighting its practical utility in large-scale recommendation scenarios.

To improve personalization, Chen, Zhang, Zhang, & Liu (2022) proposed Aspect-based Collaborative Filtering (AXCF), which fuses collaborative filtering with aspect-level sentiment analysis and excels in cold-start scenarios. However, AXCF struggled when users prioritized multiple attributes (e.g., service and ambiance), underscoring the difficulty of balancing complex, multi-dimensional preferences.

Visualization research reinforces the need for clarity. Personalized Tag Clouds (Gedikli, Ge, & Jannach 2011) heightened satisfaction and transparency via color-coded sentiment cues but risked cognitive overload. Spatial map visualizations (Kunkel & Ziegler 2023) increased enjoyment and novelty due to positive cognitive bias, also known as "halo effect", though sometimes at the cost of higher cognitive strain. Tsai & Brusilovsky (2019) showed that concise text paired with simple graphics improved comprehension and acceptance, while user tolerance for complexity varied, suggesting visualization designs must adapt to individual cognitive loads.

Collectively, these studies highlight that effective modern recommendation systems must merge robust, hybrid sentiment analysis with intuitive, adaptable visualizations to deliver accurate, interpretable, and user-satisfying recommendations.

Proposed Method

Traditional recommendation systems primarily utilize quantitative data, (i.e., user ratings, popularity metrics, etc.), often neglecting qualitative user-generated content that contains nuanced preferences. By integrating behavioral analytics, advanced sentiment analysis, and personalized qualitative filtering, our approach provides recommendations that align closely with individual user sentiments and context-specific "vibes" (e.g., ambiance, energy, aesthetics). Leveraging hybrid sentiment analysis combining lexicon-based and machine learning-driven techniques further enhances recommendation accuracy, transparency, and personalization beyond standard collaborative filtering or content-based systems. Moreover, research by Cremonesi et al. (2010) shows that simpler recommendation models can often perform comparably to more complex, state-of-the-art approaches. This suggests that achieving strong recommendation performance does not always require sophisticated or computationally intensive algorithms. In this context, our relatively simple model offers a favorable trade-off—delivering effective results with greater efficiency, interpretability, and ease of deployment. While it may not outperform the most advanced models, its comparable performance underscores its practicality for real-world use cases.

Two-Tier Sentiment Analysis

Our recommendation system utilizes a hybrid sentiment analysis framework combining VADER and Google Cloud Natural Language API. The first-tier leverages VADER, a lexicon-based sentiment analysis tool that processes cleaned review text using its **polarity_scores()** function. This function outputs sentiment values including positive, neutral, negative, and compound scores. Building on VADER's existing lexicon, we developed a custom "vibes" dictionary tailored specifically to restaurant reviews. Each term in the dictionary is assigned a sentiment score ranging from –4.0 to 4.0 reflecting how VADER rates them in its lexicon, and grouped into different "vibes", qualitative categories such as ambiance, energy, aesthetics, food, and service.

To personalize sentiment analysis, each review's sentiment score is adjusted based on user-defined preferences and the frequency of vibe-related keywords present in the text. This adjusted sentiment score for a recommendation item R_i is computed by:

$$R_i = avg_{uws}\left(C_s \times P_w \times \frac{V_c}{\max(1, \sum V_c)}\right)$$

where:

- R_i is the aggregated recommendation score for item i
- avg_{uws} is the average of all weighted scores across reviews
- C_s is the sentiment base score (e.g., VADER compound)
- P_w is the user's weighted preference
- V_c is the frequency of vibe category terms in a review

This scoring formula captures both the intensity and relevance of sentiment by accounting how often a vibe appears in a review (via V_c), and how important that vibe is to the user (via P_w).

After computing scores for all items, the system generates the Top-n recommendations by ranking them in descending order:

$$Top\text{-}n = \{R_{(i)}, R_{(i+1)}, \ldots, R_{(n)}\} \quad \text{s.t.} \quad R_{(i)} \geq R_{(i+1)} \geq \cdots \geq R_{(n)}$$

This two-step process—adjusted scoring and rank-based selection—ensures that the resulting recommendations are both contextually relevant and personalized. Moreover, averaging across multiple reviews enhances robustness at the item level (e.g., entire restaurant) rather than relying on single-review sentiment.

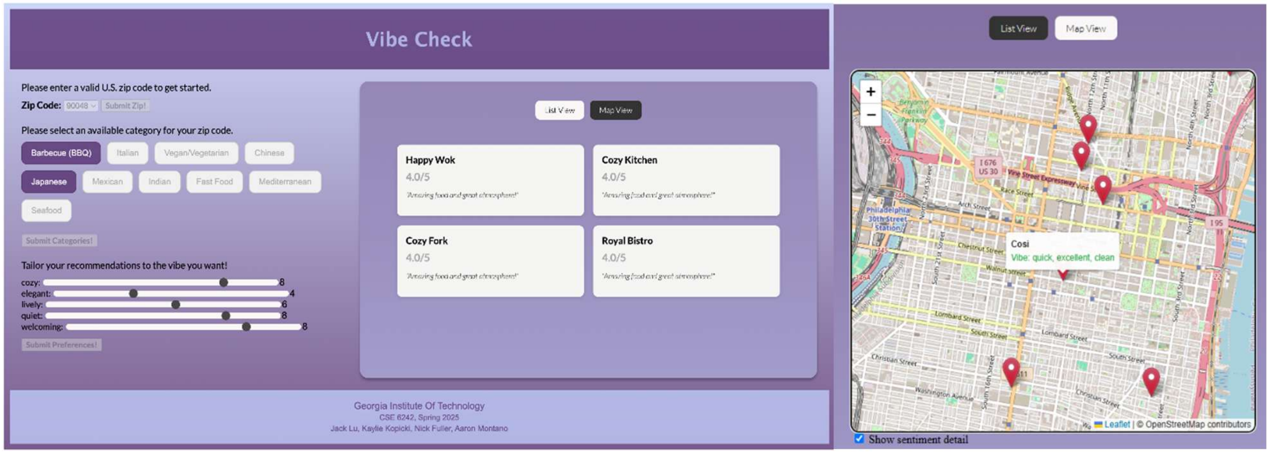
Sentiment classification is finalized using threshold cutoffs: scores above 0.05 are labeled positive, below –0.05 as negative, and scores in between as neutral.

The second tier introduces Google Cloud NLP as a complementary validation layer. While less customizable, this machine learning-based service acts as a safeguard for VADER classifications—especially when reviews are initially labeled as neutral. Although VADER's threshold reduces the risk of missing sentiment labels, its performance with a custom-built restaurant lexicon remains uncertain. This additional layer mitigates that uncertainty, providing greater confidence in edge cases and a dependable fallback mechanism when the primary model falls short.

User Interface and Visualization

To enhance usability, we developed intuitive UI features including "Vibe Weight Sliders" allowing dynamic user input on the relative importance of specific "vibes." The recommendation system then aggregates user sentiment preferences into a personalized ranking.

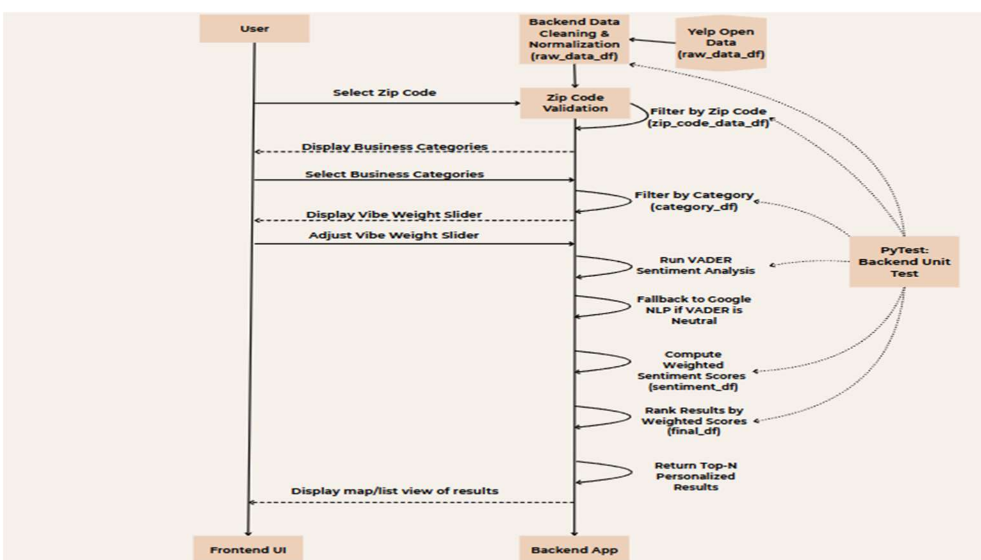
Visualization strategies are integrated to ensure recommendations are transparent and accessible. An interactive map interface displays each venue with a red marker; when hovered over, a tooltip appears showing the venue's name and top three positive "vibe" keywords in green, indicating positive sentiment derived from sentiment analysis. Clicking a marker opens a detailed pop-up card containing the business name, address, and a review snippet, along with a one-line explanation ("Recommended for its [vibe-attributes]") based on VADER's sentiment analysis to enhance transparency and build user trust. A toggle at the top of the interface allows users to switch between a simplified grid-based list of restaurants and a spatial map view, helping reduce the cognitive overload risks.



Vibe Check employs a structured data pipeline and responsive user-interaction workflow to deliver personalized venue recommendations. The system uses the publicly available Yelp Open Dataset, comprising over 8 million reviews and 200,000+ businesses across U.S. cities. Structured in JSON format, the dataset requires no scraping and ensures consistency in preprocessing. Raw Yelp data is cleaned and structured into **raw_data_df**, then filtered by user-provided zip codes (**zip_code_data_df**) and selected business categories (**category_df**). These server-side operations minimize computational overhead and improve responsiveness. Filtered reviews undergo sentiment analysis to generate a scored dataset (**sentiment_df**). Users then apply custom attributes weightings to rank and refine final recommendations (**final_df**).

User input—including zip code, category, and vibe preferences—is captured through the frontend and passed to the backend, which performs all transformations and returns real-time, ranked results. This separation reduces front-end complexity and supports scalable performance.

To ensure system reliability, PyTest is integrated throughout the development cycle. It validates core logic—including sentiment weighting, filtering, and ranking—through automated unit tests. This guarantees no failing tests at deployment, supporting robust functionality and smooth feature iteration.



Evaluation

We decided to evaluate our model via the number of identified attributes and the “vibes” tags in the results over methods such as NDCG@K, to focus on accuracy over rank awareness. Due to the different constraints and nature of this project, we are limiting our scope to only address the accuracy (such as the 3 vibes users will see) over the entire ranking awareness (such as the list of results when searching for vibes) to prevent scope creep.

Using Excel's newer text extraction functions, we extracted both the number of identified attributes and a list of “pos words” (positive words) generated by the VADER model. These words will be used as “vibe” tags in the front-end visualization, thus, this directly evaluates what users will see on the UI, the vibes, making it a functional test on both front and back end and ensure the output reflects users' expectations for vibes. To evaluate the model’s performance, we manually reviewed the original (pre-cleaned) reviews to determine whether each identified “vibe” and VADER sentiment classification was correct (scored as 1) or incorrect (scored as 0).

For Precision@3 Vibes, we examined the top three words from the “pos words” list. If any duplicates appeared within the top three, we counted them only once. If there were fewer than three words, we still evaluated accuracy based on the available one or two and divided the result by 3 to maintain consistency. With a sample size of 213, the total number of correctly identified “vibes” was 499 out of 639 possible labels.

For the VADER sentiment evaluation, we calculated accuracy by comparing the number of correct sentiment classifications to the total number of attributes evaluated. This allowed us to assess the precision of the qualitative sentiment labels generated by the model.

W10 `=VALUE(LEFT(TEXTAFTER(M10, "'service': "), FIND("}", TEXTAFTER(M10, "'service': ")) - 1))`

	I	L	M	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	text	pos words	vader sentiment	# of identified ambiance	# of correct ambiance	# of identified energy	# of correct energy	# of identified aesthetic	# of correct aesthetic	# of identified food	# of correct food	# of identified service	# of correct service	# of identified VADER sentiment	# of correct VADER sentiment	# of correctly identified Vibes	Precision@3 Vibes per row	Precision@k VADER Sentiment per row
7	and they have extensive vegan and gluten free options if you lean that way.																	
8	Their prices were very good, and the lotus cake I tried was tasty. I can't speak for their other pastries, so I'll have to stop by again sometime.	['tasty']	['Positive', 3.612, 0.3612, {'ambiance': 0, 'energy': 0, 'aesthetic': 0, 'food': 1, 'service': 0}]	0	0	0	0	0	0	1	1	0	0	1	1	1	33%	100%
9	Had their vegan (with real mayo) BLT-A and one of their salads. Both were very tasty. Much bigger portions than we had expected, though not overwhelming.	['tasty']	['Positive', 3.612, 0.3612, {'ambiance': 0, 'energy': 0, 'aesthetic': 0, 'food': 1, 'service': 0}]	0	0	0	0	0	0	1	1	0	0	1	1	1	33%	100%
10	BEST SANDWICHES EVER. The gluten free bread options alone would keep me coming back but the sandwiches themselves are incredible. Love the sides of potato salad and plan to try a salad soon. The online order and window pickup makes it the perfect lunchtime order spot. Knocked off 1 star because I'm not a huge fan of their latte flavors and wouldn't come here for only a drink, but the quality of food makes up for it.	['perfect']	['Positive', 4.019, 0.4019, {'ambiance': 0, 'energy': 0, 'aesthetic': 0, 'food': 1, 'service': 0}]	0	0	0	0	0	0	1	1	0	0	1	1	1	33%	100%

Precision@3 “Vibes” (213 sample size)	Total # of Identified VADER Sentiment (177 sample size)	Total # of Correct VADER Sentiment (177 sample size)	VADER Sentiment Accuracy (177 sample size)
78%	353	567	62%

Additionally, we maintained a log to track how often Google Cloud NLP was used as a fallback when the VADER model returned a neutral sentiment. Out of 6,367 reviews labeled as neutral by VADER, only 7 were successfully reclassified by Google NLP as either positive or negative. This outcome suggests that Google NLP offers minimal improvement in handling neutral classifications—particularly in the context of restaurant reviews. While Google NLP is currently invoked alongside VADER for every review, mostly to support exploratory analysis and comparative evaluation, this parallel approach may be inefficient and unnecessarily costly given its limited added value.

Given VADER's effectiveness, transparency, and efficiency, it remains the preferred primary sentiment engine. To reduce unnecessary resource use, we propose a conditional fallback strategy: instead of triggering Google NLP for all neutral results, future iterations could activate it only when the proportion of neutral reviews exceeds a defined threshold. This cutoff could be determined through a pilot study or internal analysis. Such a strategy ensures Google NLP is used only when it adds value, improving efficiency without sacrificing accuracy.

Conclusions

While Vibe Check effectively delivers sentiment-driven, personalized restaurant recommendations, several limitations remain. The system is currently limited to restaurant data from select metropolitan areas, reducing geographic and categorical diversity. Our vibes-attribute lexicon, developed in-house due to the absence of an existing restaurant-specific dictionary, offers limited coverage and may miss emerging or nuanced terms. Application architecture is split between data preprocessing and visualization, driven by technical constraints and Google NLP API quota limitations. To address this, Yelp data was preprocessed into a static CSV placed in a Dockerized container. Despite these constraints, the model achieved a Precision@3 of 78% for vibes tagging and VADER sentiment accuracy of 62%, confirming its ability to extract emotionally relevant insights from user reviews and accurately map them to user preferences.

Future iterations can enhance recommendation relevance by incorporating a penalty system for negative sentiment terms and expanding the lexicon using additional annotated data. Introducing negative vibes, expanding the zip code and venue scope, and validating outputs through user feedback or a vibe relevance testing interface could further strengthen the system. Additionally, conditional use of Google NLP, triggered only when neutral labels exceed a preset threshold, would improve both cost efficiency and accuracy. Finally, for production-scale deployment, a paid NLP tier may be considered. Together, these refinements will improve the robustness, transparency, and scalability of emotionally intelligent recommendations.

All team members made equal contributions

References

1. Chen, T., Zhang, J., Zhang, Y., & Liu, Y. (2022). AXCF: Aspect-based collaborative filtering for explainable recommendations. Expert Systems with Applications, 202(117234), 1-11. https://doi.org/10.1016/j.eswa.2022.117234
2. Choi, D. (2021). Numerical bias in recommender systems. Journal of Information Technology & Decision Making, 20(4), 1093–1120.
3. Cohen, M., & Rozenblat, C. (2015). Recommendation systems and mobility: A bibliometric approach. Procedia Computer Science, 52, 456-461.
4. Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. Proceedings of the fourth ACM conference on Recommender Systems (RecSys '10), 39–46.
5. Durbach, I. N., & Montibeller, G. (2019). Behavioral analytics: Harnessing behavioral science and data analytics for managerial decision making. European Journal of Operational Research, 278(3), 915-931.
6. Gedikli, F., Ge, M., & Jannach, D. (2011). Understanding recommendations by reading the clouds. Proceedings of the International Conference on Electronic Commerce and Web Technologies (EC-Web 2011), 196-208.
7. Kunkel, J., & Ziegler, J. (2023). A comparative study of item space visualizations for recommender systems. User Modeling and User-Adapted Interaction, 33(1), 123–159.
8. Moldovan, D. (2021). Majority vote ensemble with VADER: Sentiment analysis of product reviews using multiple sentiment tools. Computers, 10(7), 89. https://doi.org/10.3390/computers10070089
9. Nagarnaik, P., & Thomas, A. (2015). Survey on recommendation system methods. IEEE Sponsored 2nd International Conference on Electronics and Communication Systems (ICECS), 1603–1608.
10. Opara, E. C. (2022). Cloud-based machine learning and sentiment analysis (Master's thesis). California State University, Sacramento.
11. Tsai, C.-H., & Brusilovsky, P. (2019). Visualization of explanations in recommender systems. Journal of Information Visualization, 18(1), 119-134.
12. Zha, H., He, X., Ding, C., Gu, M., & Simon, H. D. (2001). Bipartite graph partitioning and data clustering. Proceedings of the tenth international conference on Information and Knowledge Management (CIKM '01), 25–32.
13. Zaki, M. J., & Hsiao, C. J. (2005). CHARM: An efficient algorithm for closed itemset mining. Proceedings of the SIAM International Conference on Data Mining (SDM), 34–43.